# FERRAMENTA WEB DE SUPORTE A AVALIAÇÃO DE SOFTWARE COM A METODOLOGIA CERTICS

Acadêmico: Vinícius Ferneda de Lima
vinicius.ferneda@gmail.com

Orientador: Prof. Everaldo Artur Grahl
everaldo.grahl@gmail.com

FURB
UNIVERSIDADE DE BLUMENAU

# Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Requisitos
- Especificação
- Implementação
- Operacionalidade da implementação
- Resultados e discussões
- Conclusões e sugestões

# Introdução

- Esta ferramenta visa apoiar a avaliação de um consultor sobre um software da organização solicitante.
- Permite aos professores da área de qualidade de software uma opção para aulas práticas.
- São restritas as soluções para adoção da metodologia CERTICS.

# Objetivos

- O objetivo geral deste trabalho é desenvolver uma ferramenta web que possa apoiar a avaliação de software usando a metodologia CERTICS.
- Os outros objetivos são:
  - disponibilizar um ambiente que permita configurar a metodologia de avaliação da CERTICS;
  - agilizar o cálculo dos resultados de uma avaliação e possibilitar a emissão de relatórios detalhados com a consequente resultância das avaliações do software;
  - permitir que os avaliadores consigam avaliar as evidências registradas pela empresa que está sendo analisada.

# Fundamentação Teórica

- O que é CERTICS?
  - é uma certificação voluntária que identifica, credencia e diferencia software resultante de desenvolvimento e inovação tecnológica realizados no País.
- Foi criada em 2013 pelo Ministério da Ciência, Tecnologia e Inovação (MCTI) como parte do programa TI Maior.

# Fundamentação teórica

- A certificação serve de instrumento às organizações que buscam qualificação em compras públicas a partir da introdução de margem de preferência a detentores de tecnologia desenvolvida no País.
- Certifica o software de Organizações desenvolvedoras instaladas no território nacional.

# Fundamentação teórica

- Quem se beneficia?
  - Empresas desenvolvedoras de Software.
  - Compradores responsáveis por licitações.
  - Entidades compradoras.
- A metodologia CERTICS é considerada:
  - Inclusiva.
  - Ágil.
  - Acessível.

# Fundamentação teórica

- A metodologia segue as seguintes diretrizes:
  - A avaliação é do software, não da empresa, e é baseada na análise dos processos utilizados no software.
  - A metodologia é baseada na Norma ABNT NBR ISO/IEC 15504 para avaliação de processo.
  - Um novo conceito demanda um novo modelo de referência e um novo método para avaliação.

# Fundamentação teórica

- A avaliação é baseada em evidências que demonstram o grau de atendimento de resultados esperados em quatro áreas de competências.
- As quatro áreas de competências são as seguintes:
  - Desenvolvimento Tecnológico (DES).
  - Gestão de Tecnologia (TEC).
  - Gestão de Negócios (GNE).
  - Melhoria Continua (MEC).

# Fundamentação teórica

# Fundamentação Teórica

- O resultado de uma avaliação na metodologia CERTICS é definido a partir da pontuação de cada resultado esperado. A pontuação é classificada da seguinte forma:
  - completamente atendido (*fully* – F);
  - largamente atendido (*largely* – L);
  - parcialmente atendido (*partially* – P);
  - não atendido (*not* – N).

# Trabalhos Correlatos

- WISE: Uma Abordagem de Ferramenta de Software Para o Auxílio no Modelo de Avaliação do MPS.BR.
- Ambiente Web de Suporte ao Processo de Avaliação da Qualidade de Produtos de Software.
- CERTICSys.

# Requisitos

Baseado nos trabalhos pesquisados correlatos e na descrição da metodologia CERTICS foram identificados os seguintes requisitos:

- permitir o cadastro de usuário (Requisito Funcional – RF);
- permitir a definição de áreas de competência (RF);
- permitir a definição dos resultados esperados (RF);
- permitir o cadastro de avaliador (RF);

# Requisitos

- permitir o cadastro de organização (RF);
- permitir o cadastro de profissionais da organização (RF);
- permitir o cadastro de softwares (RF);
- permitir o cadastro de evidências (RF);
- permitir o cadastro de anexos (RF);
- permitir a definição de escalas de pontuação (RF);
- permitir a criação de uma avaliação de software (RF);

# Requisitos

- permitir o vínculo de evidências em uma avaliação (RF);
- permitir o registro de observações sobre uma avaliação (RF);
- permitir o cálculo do resultado de uma avaliação (RF);
- permitir a geração de um gráfico de atendimento das áreas de competência (RF);
- permitir a geração de um relatório do resultado final da avaliação detalhado (RF);

# Requisitos

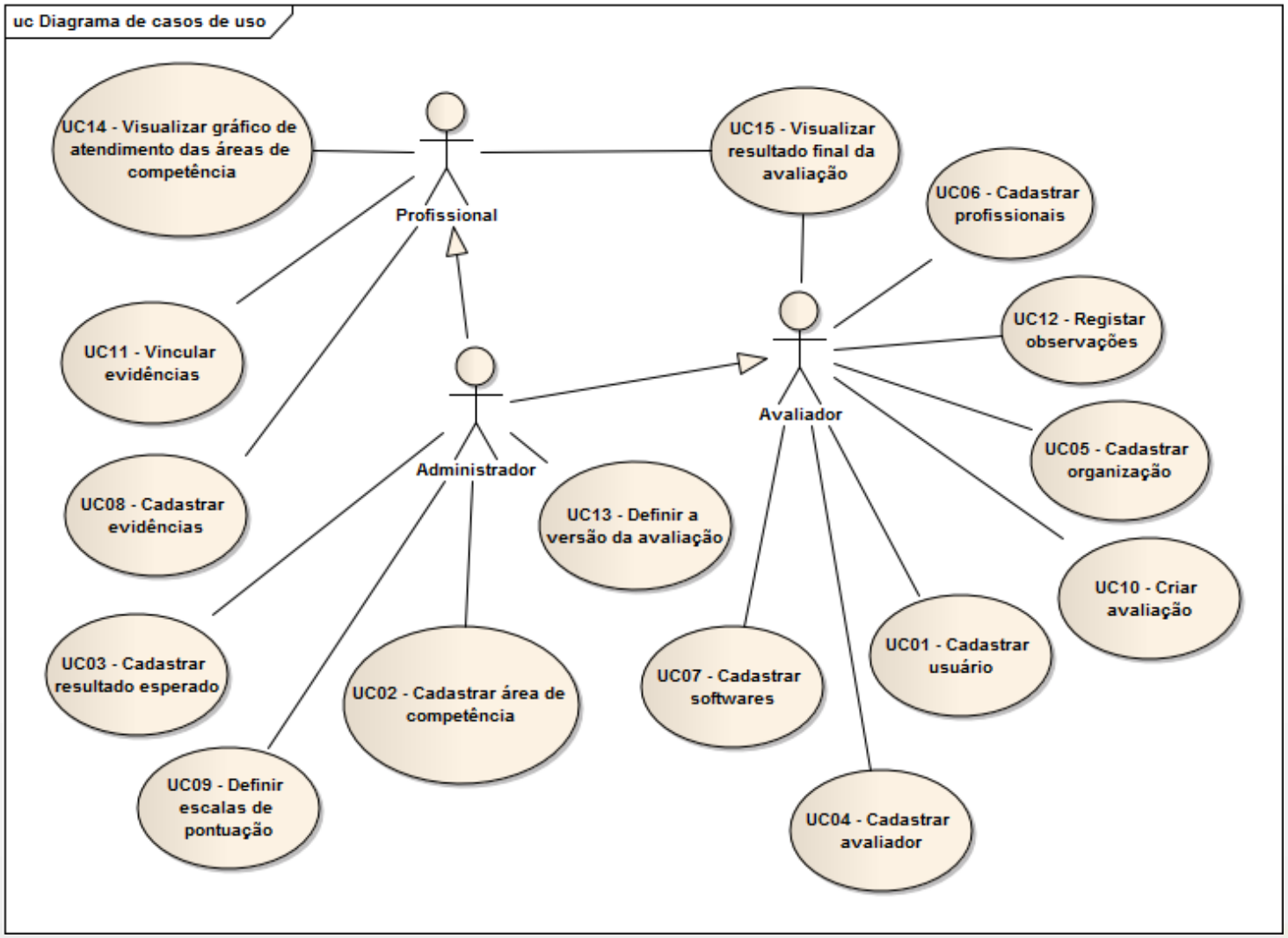
- utilizar a linguagem de programação Java 1.7 (Requisito Não-Funcional - RNF);
- utilizar o ambiente de desenvolvimento Eclipse 4.3 (RNF);
- utilizar JSF 2.0 com o *framework* PrimeFaces para que a ferramenta ofereça suporte web (RNF);
- utilizar JasperReports (RNF);
- utilizar iReport (RNF);
- utilizar Banco de dados PostgreSQL (RNF).

# Especificação

Foram utilizados os seguintes diagramas:

- Casos de uso;
- Classes;
- Atividades.
- Também foi especificado o Modelo Entidade Relacionamento (MER).

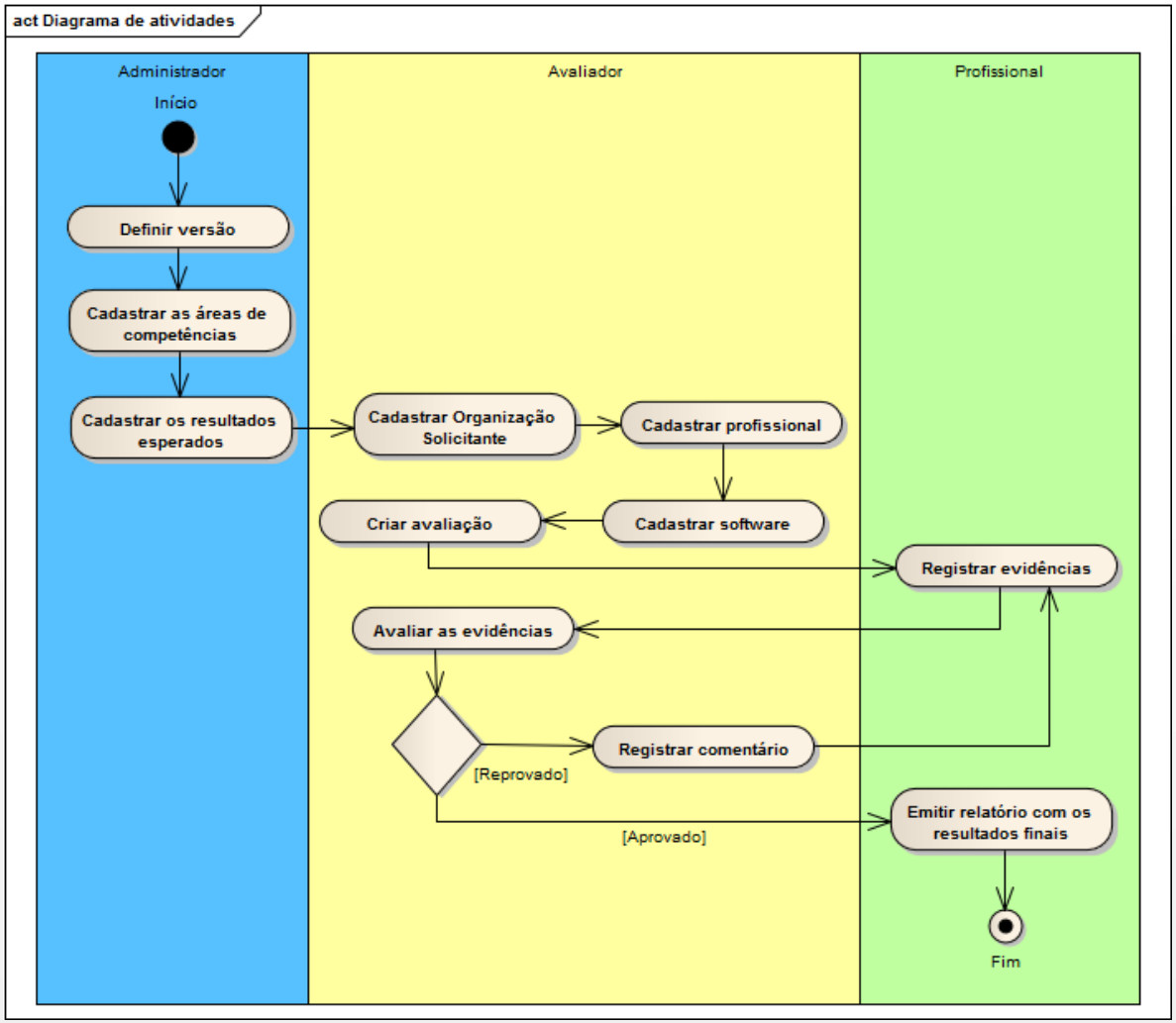
# Especificação



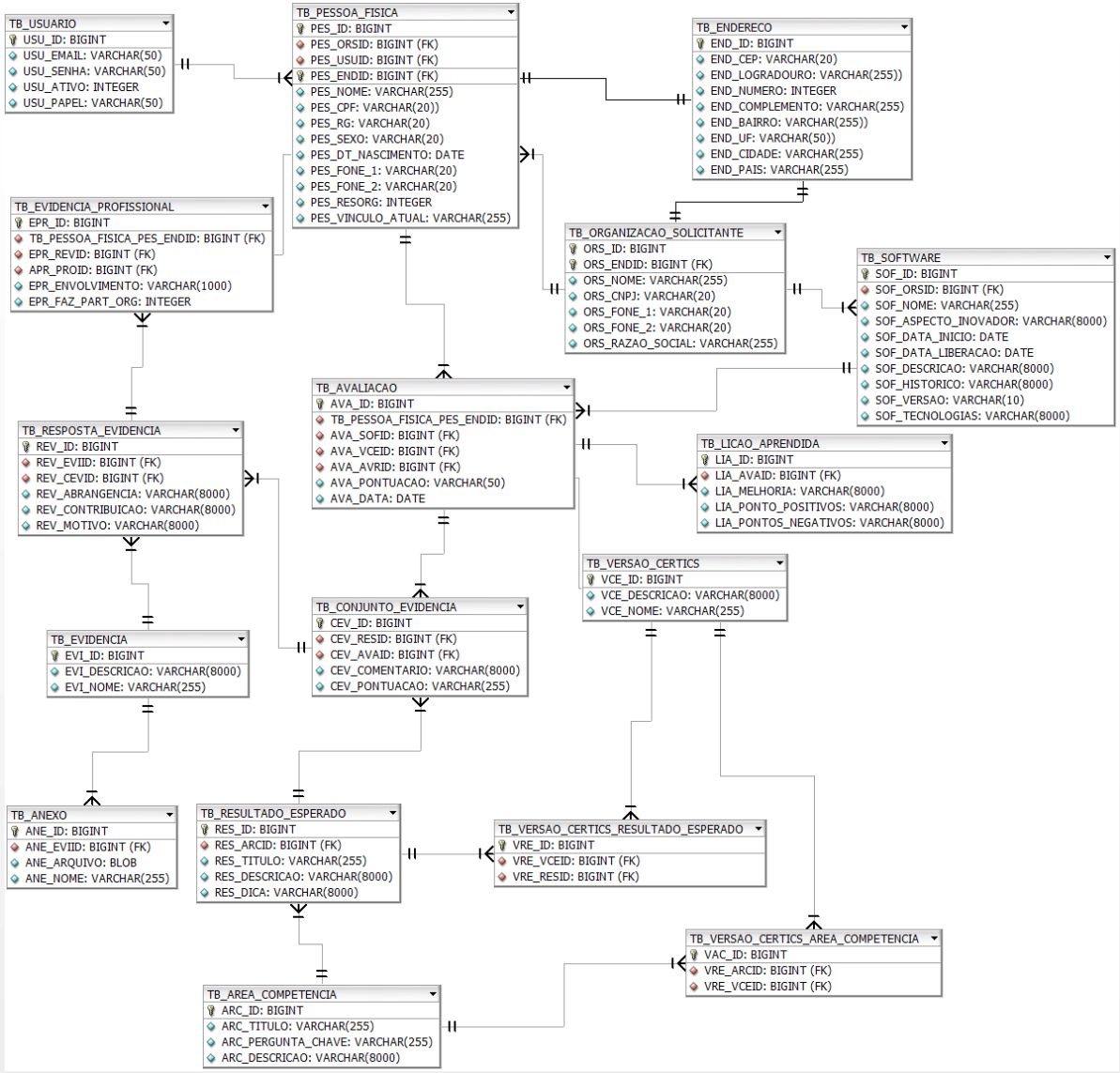
# Especificação

- Foram criados dois diagramas de classes.
- O primeiro tem por objetivo identificar as entidades são responsáveis pela lógica de negócio.
- O Segundo é referente à funcionalidade do vínculo de evidências ao software.

# Especificação



# Especificação

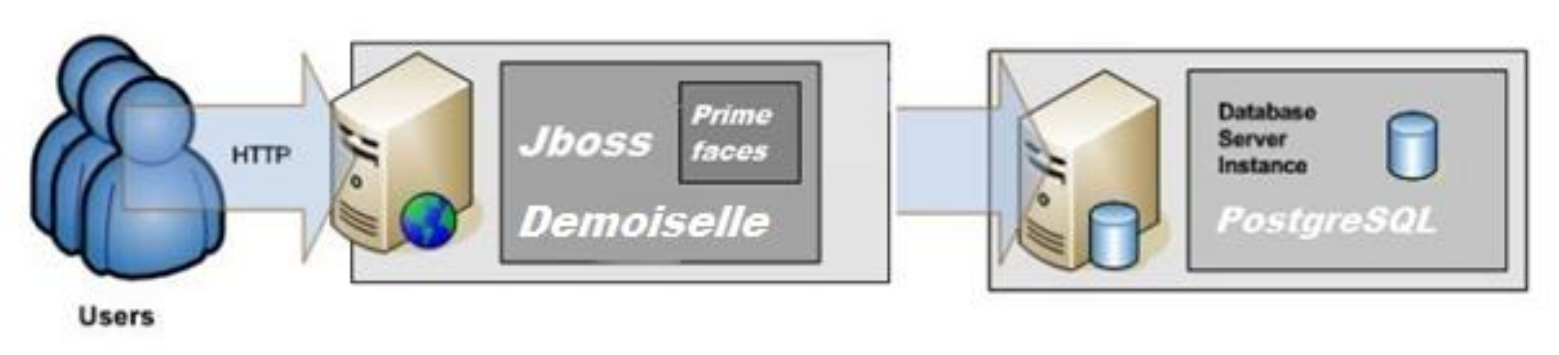



FURB
UNIVERSIDADE DE BLUMENAU

# Implementação

- Foi escolhida a linguagem JAVA na versão 7 para o desenvolvimento da ferramenta.
- Foram utilizadas as seguintes tecnologias:
  - *Framework Demoiselle;*
  - JBoss AS 7.1.1.Final;
  - PrimeFaces;
  - PostgreSQL;

# Implementação



# Implementação

- Para a construção da arquitetura da ferramenta foi seguido o padrão MVC.
- No *framework Demoiselle* este padrão é definido através da utilização dos seguintes estereótipos:
  - *@ViewController;*
  - *@BusinessController;*
  - *@PersistenceController.*

# Implementação

```
1  <p:layoutUnit id="treeRegistroEvidencias" position="west"
2  resizable="true" size="20%">
3      <p:tree id="tree" value="#{conjuntoEvidenciasEditMB.root}" var="node"
4      cache="false" selectionMode="single"
5      selection="#{conjuntoEvidenciasEditMB.selectedNode}"
6      style="width:100%;height:100%">
7
8          <p:ajax event="select" listener="#{conjuntoEvidenciasEditMB.onNodeSelect}"
           update=":formRegistroEvidencias:panelPrincipalRegistroEvidencia,
          :formRegistroEvidencias:panelPrincipalRegistroEvidenciaAvaliador,
          :formRegistroEvidencias:panelPrincipalRegistroEvidenciaAvaliado,
          :formRegistroEvidencias:panelPrincipalComentarioAvaliador,
          :formRegistroEvidencias:dtEvidencias"/>
9
10         <p:treeNode id="treeNode">
11             <h:outputText id="lblNode" value="#{node.titulo}"/>
12         </p:treeNode>
13     </p:tree>
14 </p:layoutUnit>
```

# Implementação

```
1  public TreeNode getRoot(){
2      List<AreaCompetenciaEntity> lAreaCompetencia = this.areaCompetenciaDAO.
       findByVersaoCerticsAndAvaliacaoID(this.getId(),
3      this.getBean().getVersaoCertics().getId());
4      this.root = new DefaultTreeNode("root", null);
5      for (AreaCompetenciaEntity areaCompetenciaEntity : lAreaCompetencia) {
6          TreeNode areaCompetencia = new DefaultTreeNode(new InformacoesArvore(
           areaCompetenciaEntity.getId(), areaCompetenciaEntity.getTitulo(),
           "areaCompetencia"), root);
7          for (ResultadoEsperadoEntity resultadoEsperadoEntity :
           areaCompetenciaEntity.getResultadosEsperados()) {
8              TreeNode resultadoEsperado = new DefaultTreeNode(new InformacoesArvore
               (resultadoEsperadoEntity.getId(), resultadoEsperadoEntity.getTitulo(),
                "resultadoEsperado"), areaCompetencia);
9          }
10     }
11     return this.root;
12 }
13
14 public void onNodeSelect(NodeSelectEvent event) {
15     if("resultadoEsperado".equals(((InformacoesArvore)event.getTreeNode().getData
       ()).getTipo())){
16         this.getBean().setConjuntoEvidenciasAux(this.conjuntoEvidenciasDAO.
           findByResultadoEsperadoID(((InformacoesArvore)event.getTreeNode().getData
           ()).getId()));
17     }
18 }
```

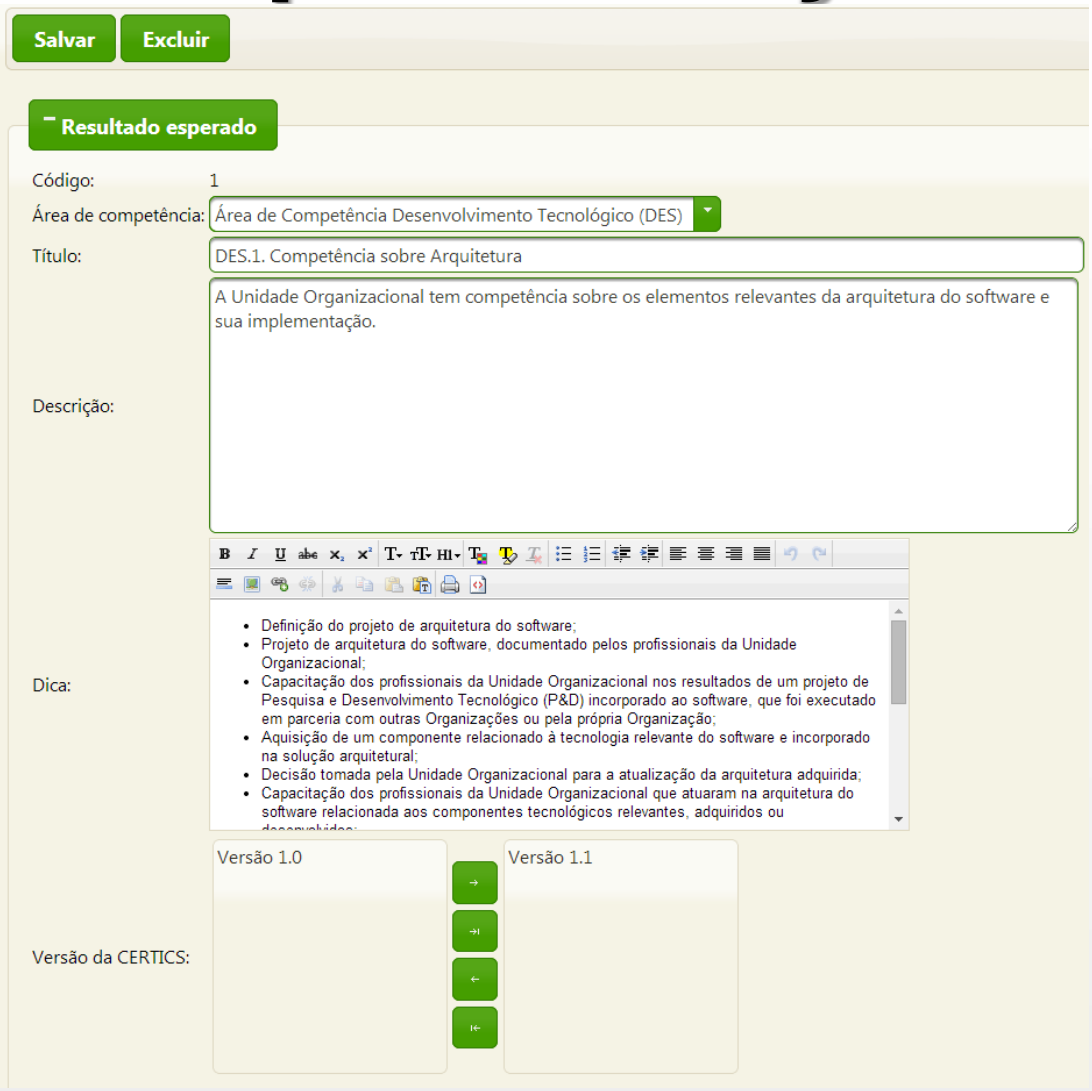
# Implementação

```
1  <h:panelGrid  id="panelPrincipalRegistroEvidenciaAvaliador" columns="2"
2  rendered="#{conjuntoEvidenciasEditMB.mostraConjuntoEvidenciaAvaliador()}">
3      <h:outputLabel value="#{messages['conjuntoEvidencias.label.pontuacao']}: "
       styleClass="text-input" />
4      <p:selectOneMenu id="pontuacao" effect="fade"
5      value="#{conjuntoEvidenciasEditMB.bean.conjuntoEvidenciasAux.pontuacao}">
6          <f:selectItems value="#{conjuntoEvidenciasEditMB.pontuacao}" />
7      </p:selectOneMenu>
8
9      <h:outputLabel value="#{messages['conjuntoEvidencias.label.comentario']}: "
       for="conjuntoEvidenciasComentario" styleClass="text-input" />
10     <p:inputTextarea id="conjuntoEvidenciasComentario"
11     value="#{conjuntoEvidenciasEditMB.bean.conjuntoEvidenciasAux.comentario}"
       title="#{messages['conjuntoEvidencias.alt.comentario']}" rows="10" cols="80"
       maxlength="8000" autoResize="false"/>
12     <p:message for="conjuntoEvidenciasComentario" />
13
14     <p:commandButton value="#{messages['button.save']}"
15     actionListener="#{conjuntoEvidenciasEditMB.insertConjuntoEvidencia()}"/>
16 </h:panelGrid>
```

# Implementação

```
1  public void insertConjuntoEvidencia(){
2      this.conjuntoEvidenciasBC.update(this.getBean().getConjuntoEvidenciasAux());
3      atualizaPontuacaoAvaliacao();
4  }
5
6  private void atualizaPontuacaoAvaliacao() {
7      int qtdRespostas = 0, completamenteAtendido = 0, largamenteAtendido = 0,
       parcialmenteAtendido = 0, naoAtendido = 0, naoRespondida = 0;
8      for (ConjuntoEvidenciasEntity conjuntoEvidenciasEntity :
       conjuntoEvidenciasDAO.findByAvaliacaoID(getId())) {
9          qtdRespostas++;
10         if(conjuntoEvidenciasEntity.getPontuacao() != null){
11             switch (conjuntoEvidenciasEntity.getPontuacao()) {
12             case F:
13                 completamenteAtendido++;
14                 break;
15             case L:
16                 largamenteAtendido++;
17                 break;
18             case P: parcialmenteAtendido++;
19                 break;
20             case N: naoAtendido++;
21                 break;
22             }
23         }else{
24             naoRespondida++;
25         }
26     }
27
28     if(naoRespondida == 0 && (parcialmenteAtendido > 0 || naoAtendido > 0)){
29         getBean().setPontuacao(EnumPontuacaoAvaliacao.REPROVADA);
30     }else if(naoRespondida == 0 && qtdRespostas == (completamenteAtendido+
       largamenteAtendido)){
31         getBean().setPontuacao(EnumPontuacaoAvaliacao.APROVADA);
32     }else{
33         getBean().setPontuacao(EnumPontuacaoAvaliacao.PENDENTE);
34     }
35     update();
36 }
```

# Operacionalidade da Implementação



# Operacionalidade da Implementação

# Operacionalidade da Implementação


FURB
UNIVERSIDADE DE BLUMENAU

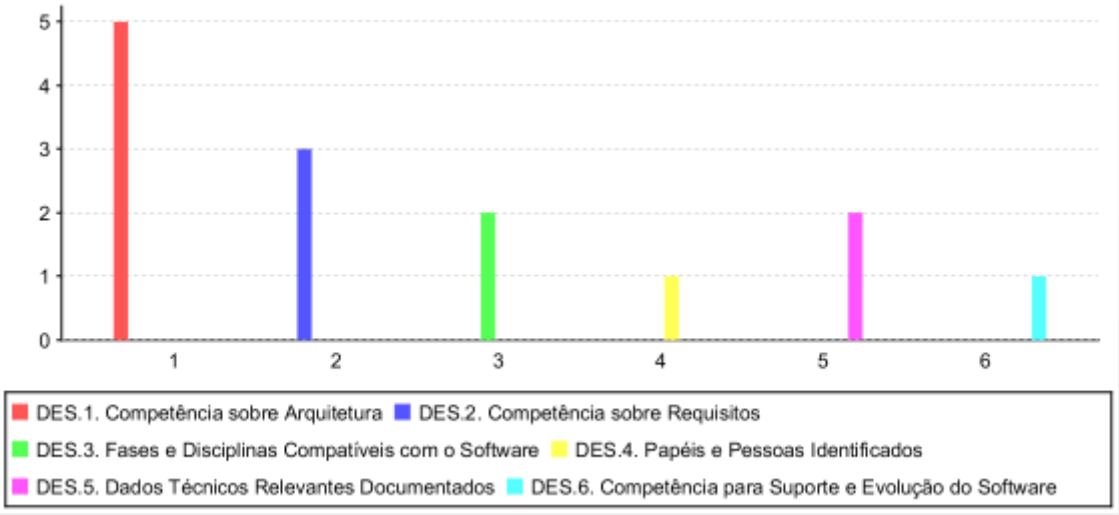
# Operacionalidade da Implementação

## Gráfico de atendimento das áreas de competência

**Área de competência:** Área de Competência Desenvolvimento Tecnológico (DES)

| Resultado esperado | Atendido? |
|---|---|
| DES.1. Competência sobre Arquitetura | Sim |
| DES.2. Competência sobre Requisitos | Sim |
| DES.3. Fases e Disciplinas Compatíveis com o Software | Sim |
| DES.4. Papéis e Pessoas Identificados | Sim |
| DES.5. Dados Técnicos Relevantes Documentados | Sim |
| DES.6. Competência para Suporte e Evolução do Software | Sim |

**Quantidade de evidências**




FURB
UNIVERSIDADE DE BLUMENAU

# Resultados e Discussões

- Tornou-se possível para o avaliador criar combinações de versões diferentes sem que seja necessário seguir as versões padrões disponibilizadas pela metodologia CERTICS.
- Tornou a avaliação mais flexível.
- São avaliadas as regras definidas pela metologia CERTICS.

# Resultados e Discussões

| FUNCIONALIDADES | TCC | Albuquerque Júnior | Borges | CERTICSys |
|---|---|---|---|---|
| Metodologia CERTICS | X | | | X |
| Ambiente web | X | | X | X |
| Evidências com anexo | X | X | X | X |
| Relatórios gerenciais | X | X | X | |
| Portal aberto para pessoas físicas | X | X | X | |
| Tutor automatizado que auxilia na identificação das evidências | | | | X |

# Conclusões e Sugestões

- O principal aspecto a ser apresentado é que a ferramenta utilizará a metodologia CERTICS para a realização da avaliação do software
- Maior facilidade ao avaliador para que as evidências sejam devidamente cadastradas e avaliadas.

# Conclusões e Sugestões

- Evita-se que o avaliador não tenha que fazer uso de outros softwares como editores de texto e planilhas de cálculo.
- Tende a diminuir o tempo gasto pelo avaliador e a organização avaliada.
- Pode-se utilizar a ferramenta para dinamizar o ensino de qualidade de software.

# Conclusões e Sugestões

- Construir um recurso que gere as dicas dos resultados esperados com base nas evidências já cadastradas na ferramenta.
- Criar um forúm que possibilite o avaliador e a organização solicitante se comunicar dentro da ferramenta.
- Disponibilizar um mecanismo que realize uma avaliação prévia automaticamente das evidências vinculadas.

# Obrigado!